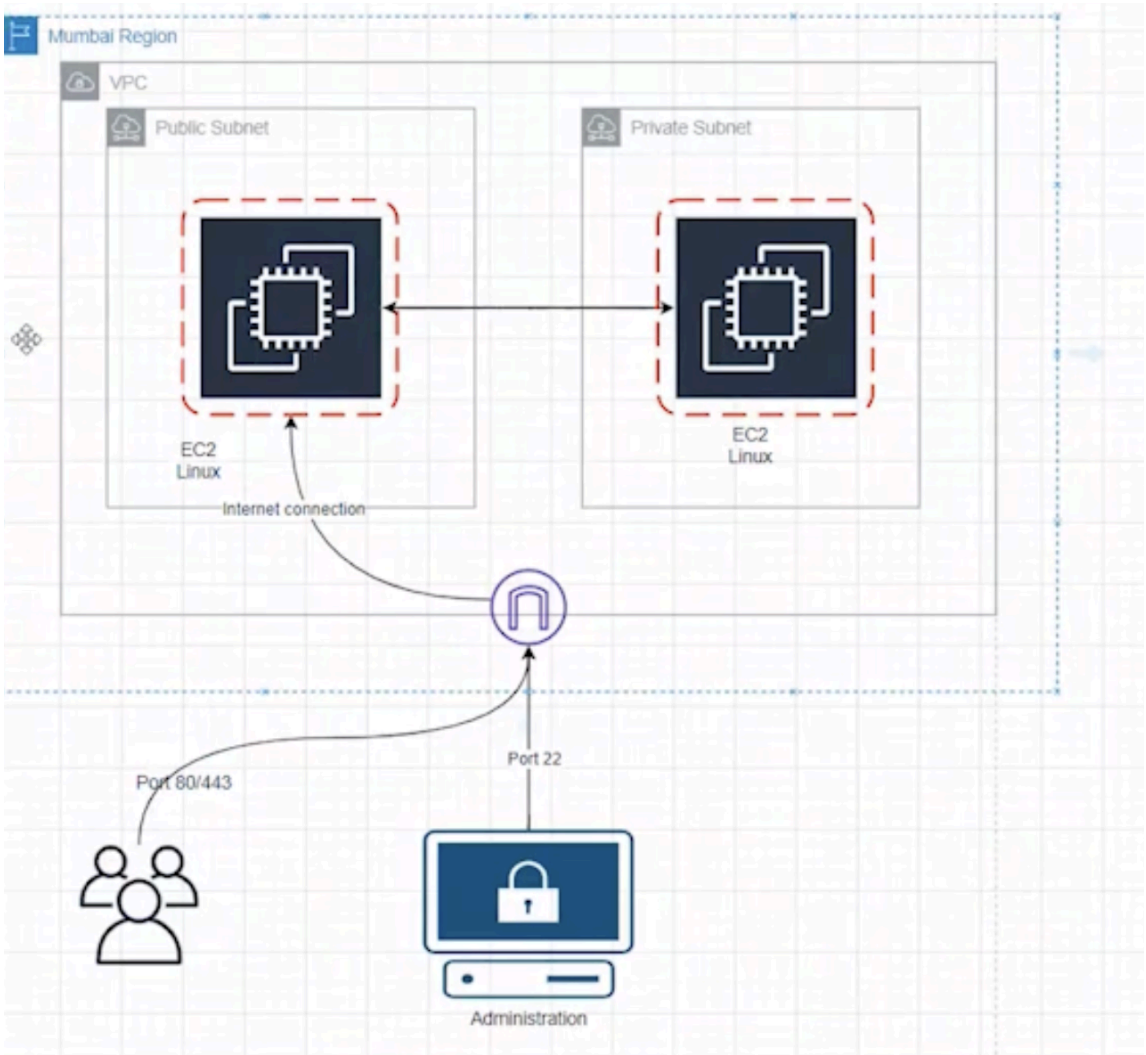
AWS VPC Hands-On: Public and Private EC2 with Internet Gateway

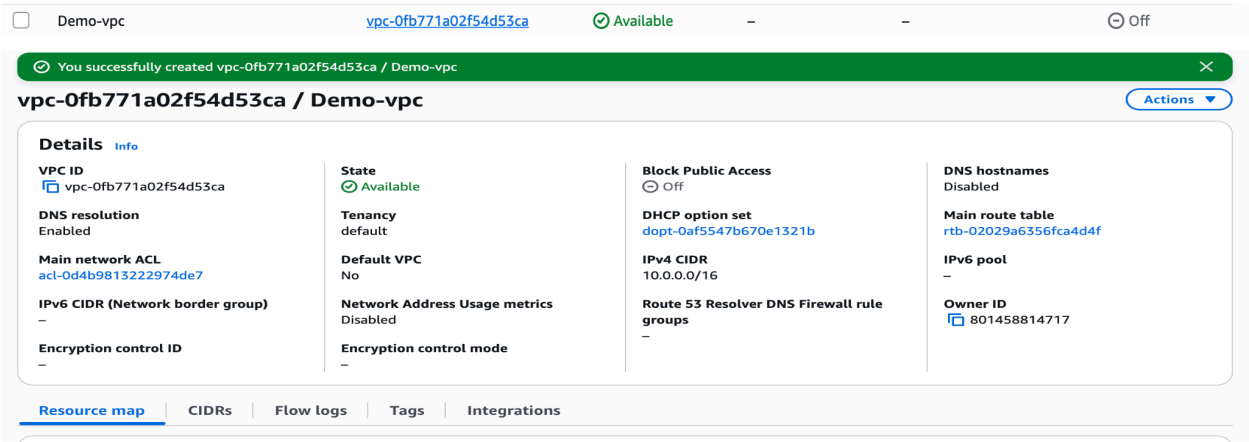
→create networking on AWS which includes AWS VPC, Subnets, Internet Gateway, Route Tables, Security Groups along with a server inside the network.

➢ TASK : created a custom AWS VPC with one public EC2 and one private EC2.
The public EC2 has internet access using an Internet Gateway, and I used it as a jump server to connect to the private EC2, which has no direct internet access

ADVANTAGES:
To allow internet access only where required and keep backend servers safe.



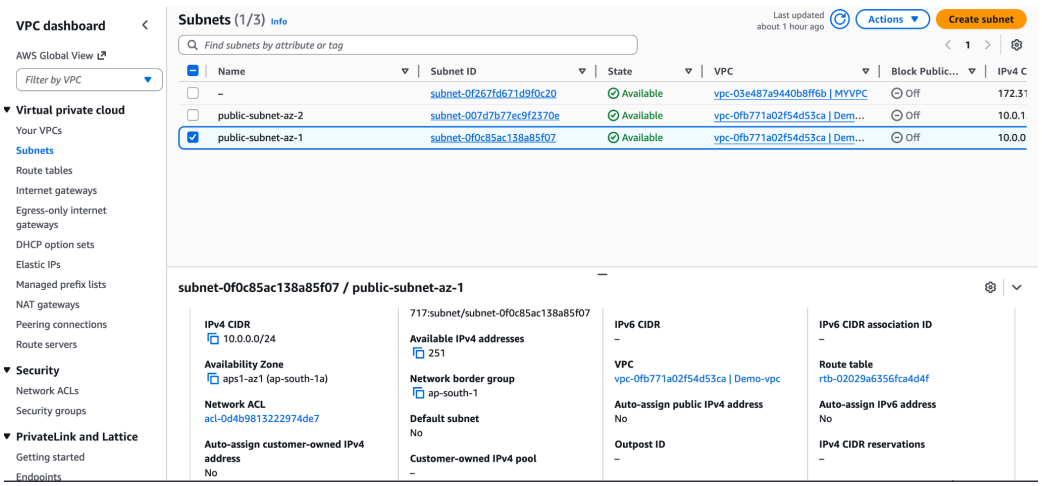
Step 1: **Select Mumbai Region**
Step 2: **Create vpc**
Name: Demo-vpc

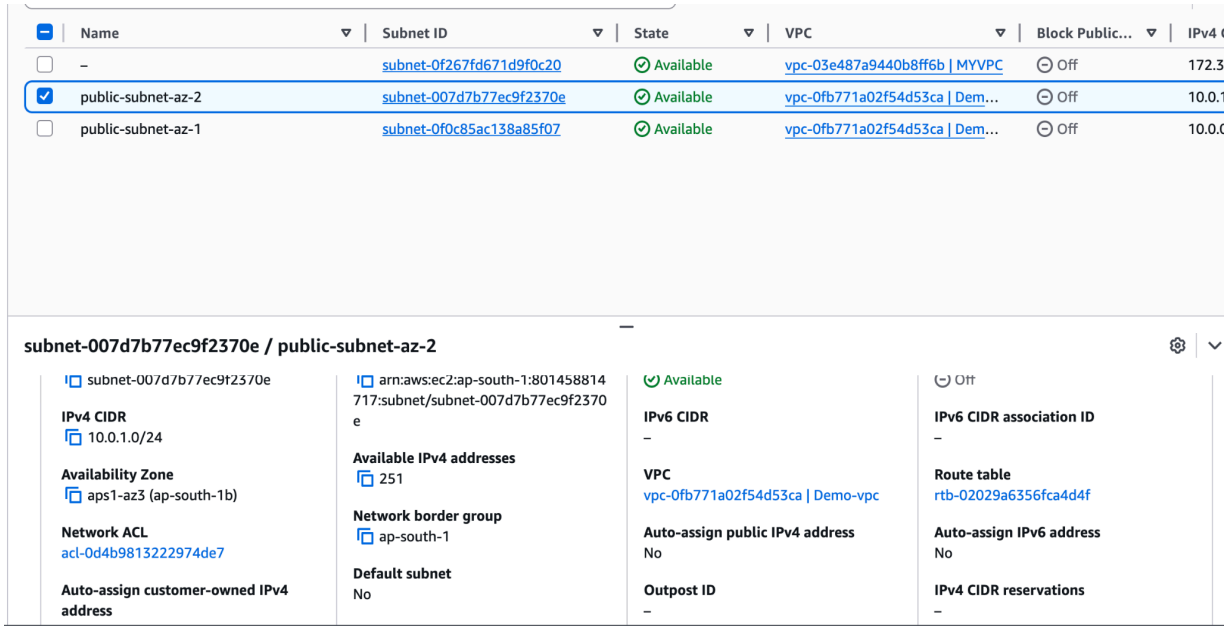


Step3 : **Create Subnets**

Go to VPC → Subnets → Create subnet
Select your VPC
Subnet name: `public-subnet-az-1`
Availability Zone: ap-south-1a
CIDR block: `10.0.0.0/24`
Add a subnet

subnet name: `private-subnet-az-2`
Availability Zone: ap-south-1b
CIDR block: `10.0.1.0/24`
Click Create subnet





Step 4: Create Internet Gateway

Go to VPC → Internet Gateways
Click Create internet gateway
Name: Demo-vpc-igw

Click Create internet gateway

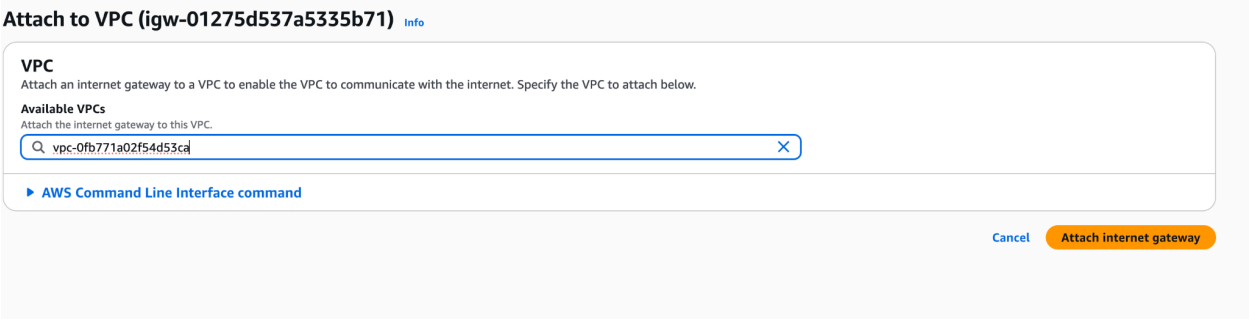
Attach Internet Gateway to VPC

Select the Internet Gateway

Click Actions → Attach to VPC

Select your VPC (Demo-vpc)

Click Attach



→ Both subnets having same route table ID. we need to change the route table .Because if we change any rule in any route table it will effect both subnets

Step 4: **create a Route table**

Go to VPC → Route Tables

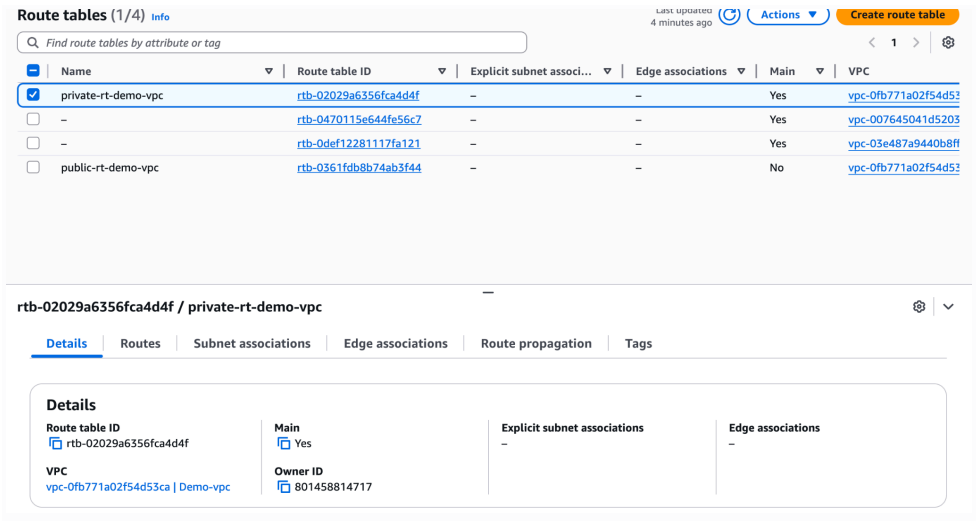
Click Create route table

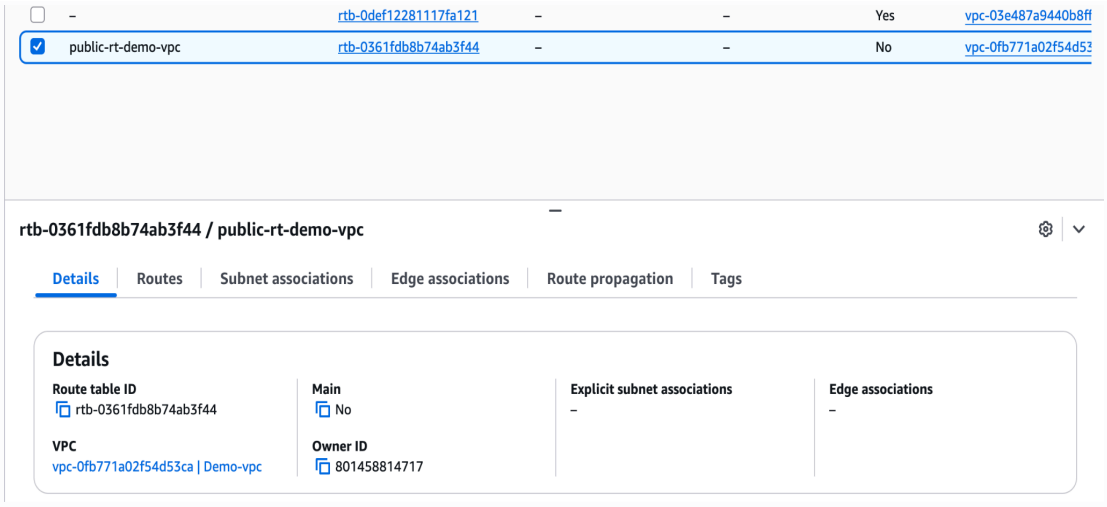
Name: public-rt-demo-vpc

Select: Demo-vpc

Create it

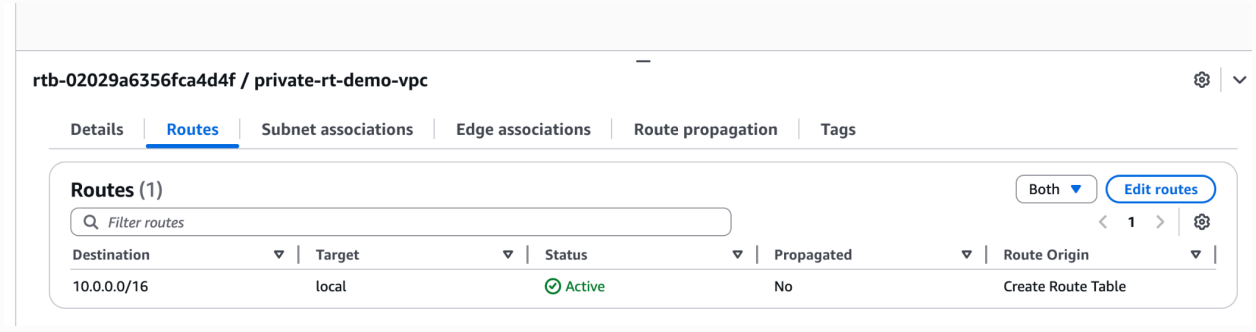
While we creating the subnet's by default one route table will create (ex route id : rtb-02029a6356fca4d4f) this is route id for both private and public subnet. Because of the we created new route table.Now in Route table we have two route table one is default another one is which we created. Default one name it has private-rt-demo-vpc





(*** now both are have different route id's)

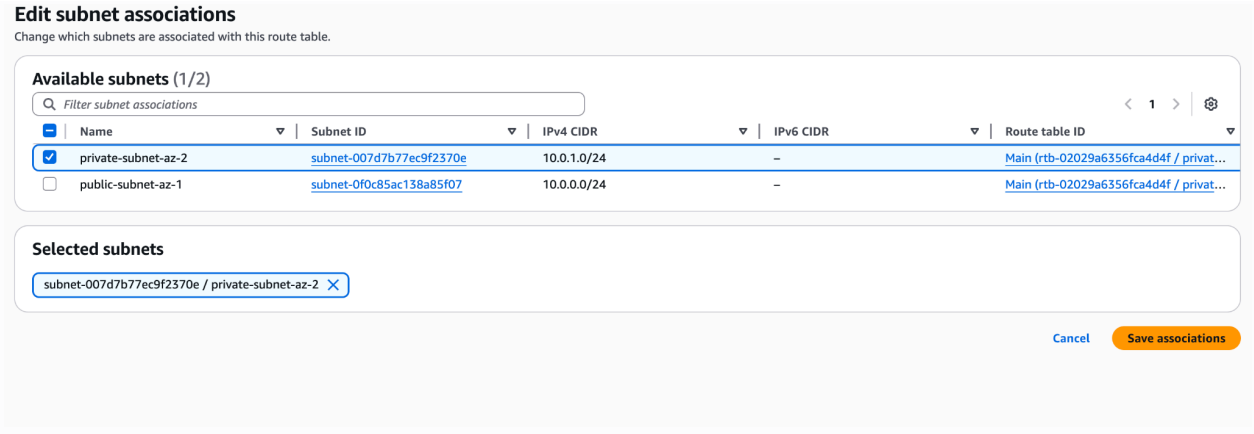
→Every route table having default IP that is 10.0.0.0/16. If vpc wants to communicate internally to router at that it will use this 10.0.0.0/16. By default every router will have this ip this is for internal communication



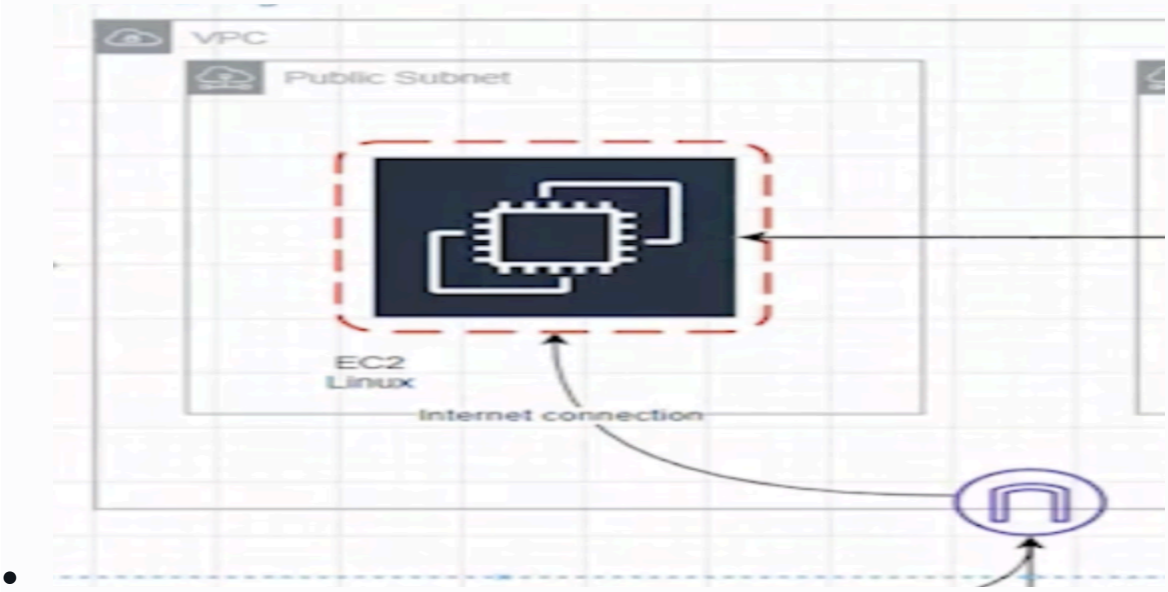
Step 5: Now private route table having Two subnet's now we need to detach one subnet

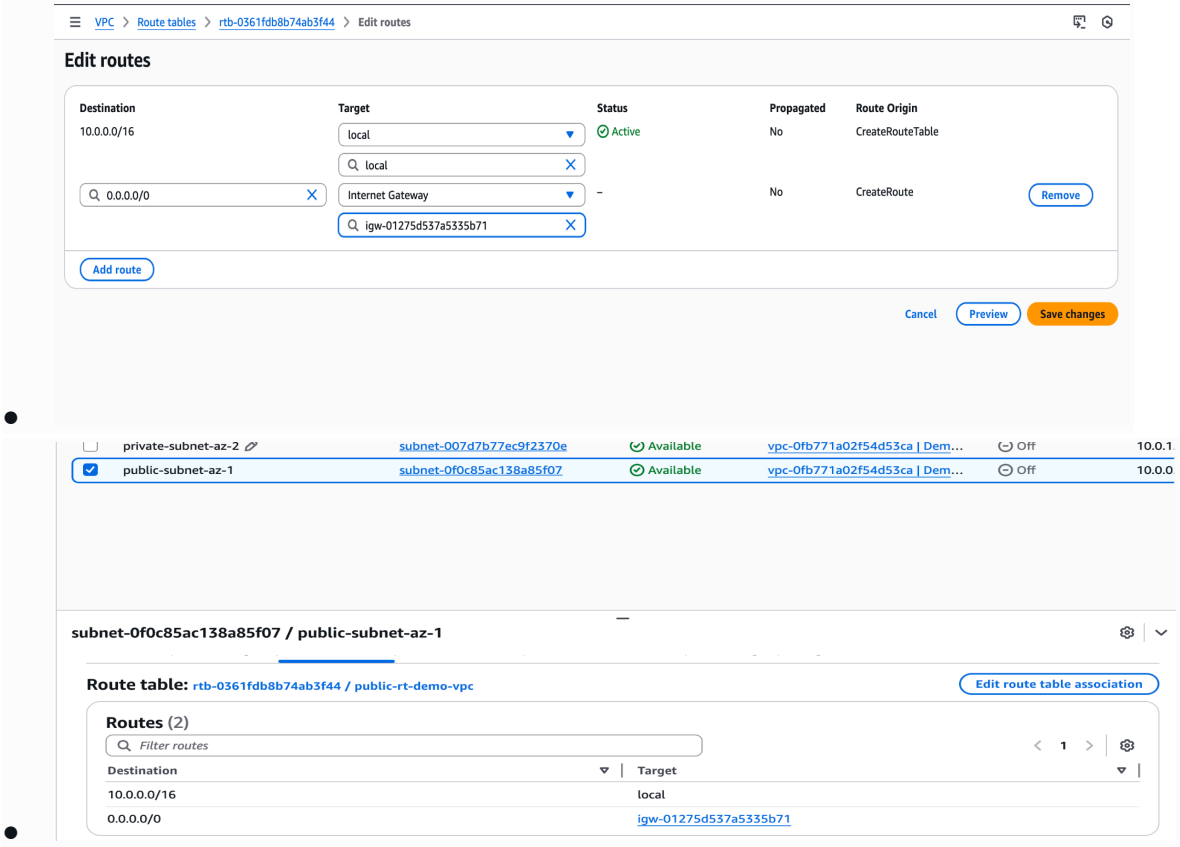
Go to private-rt-demo-vpc ,--> go to subnet association (there we have public and private subnets attached)

→for private subnet select private route and save associations



→for public subnet select public route

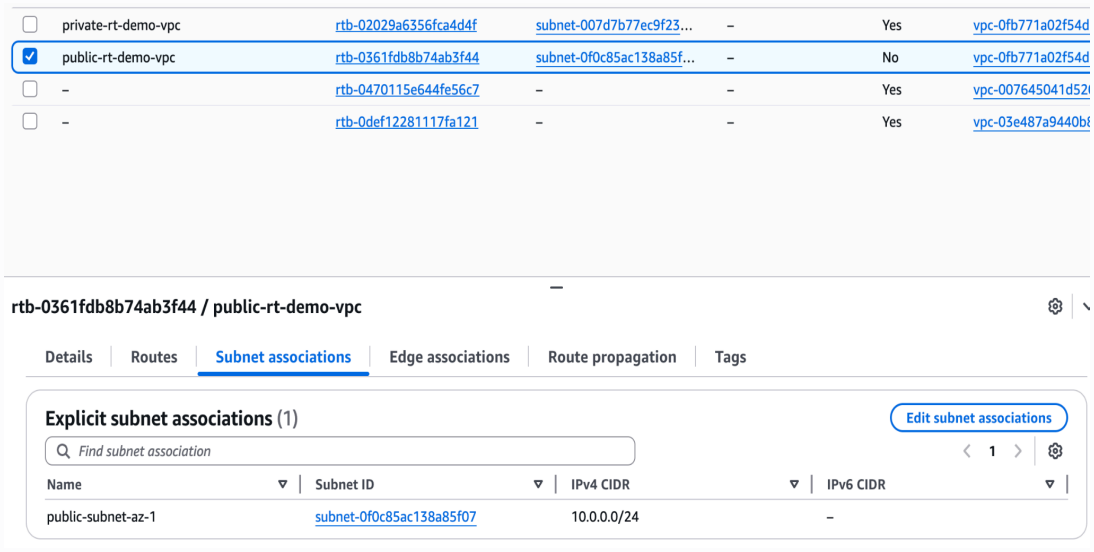
- Need internet connection for public route for that need to add the rule for public route table . for the internet connect ip is 0.0.0.0/0
- 



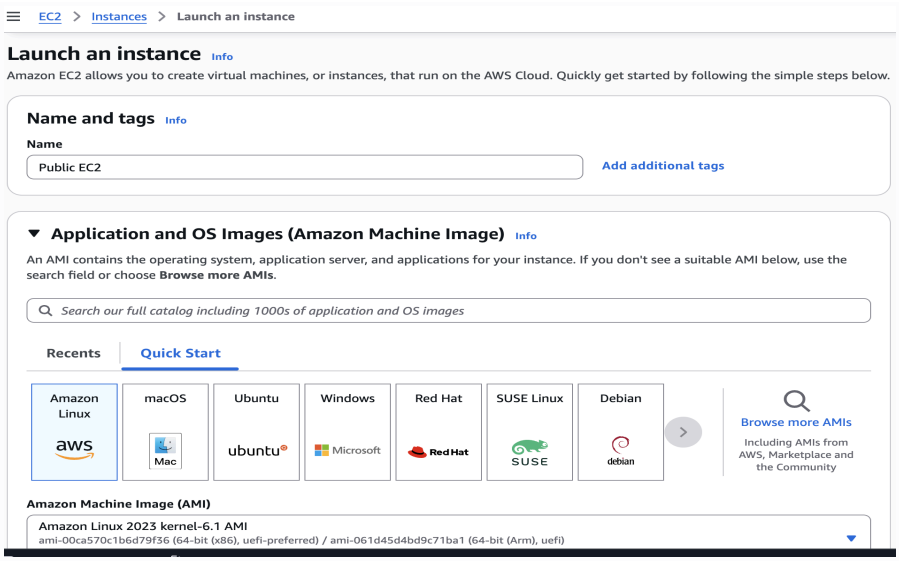
Save the changes

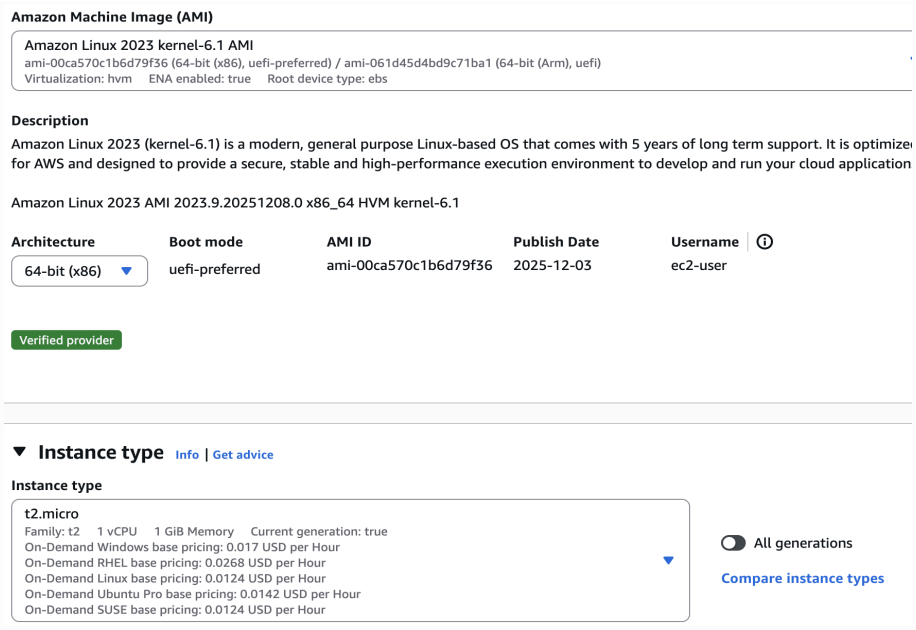
Need to edit the public route for the public subnet we need to have public route only.

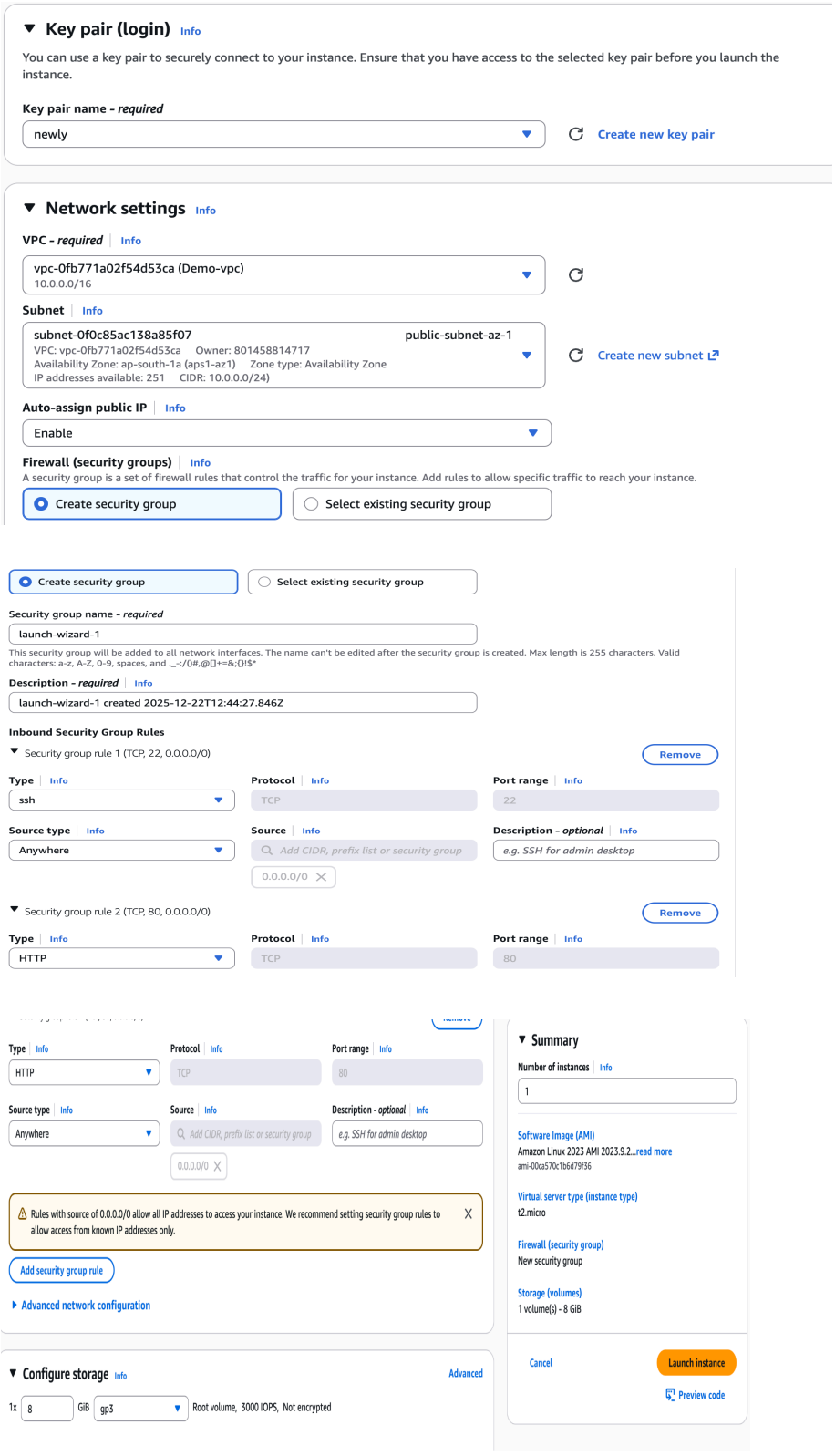
→Go to public-rt-demo-vpc ,--> go to subnet association (there we have public and private subnets attached) select the public-subnet-az-1 and save the changes



Step 6: create an ec2 instances name it has **Public EC2**

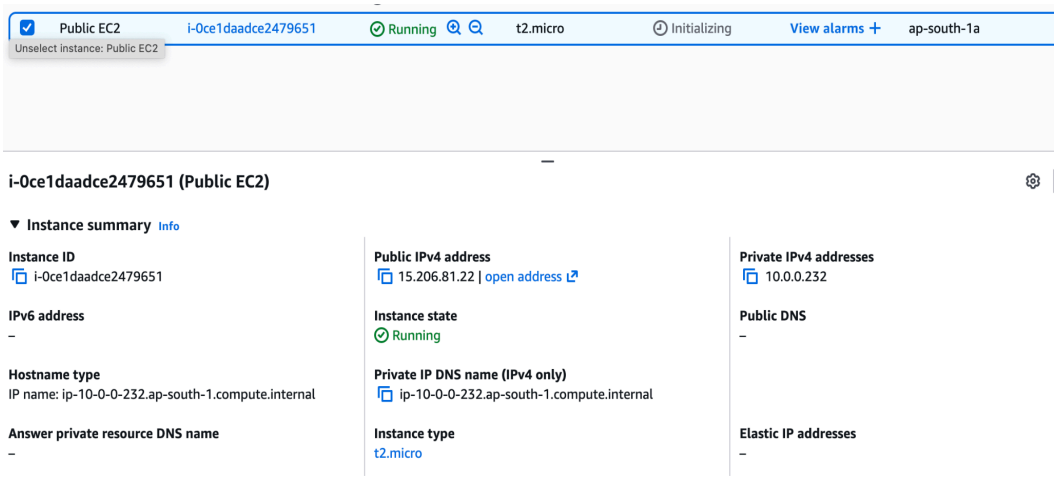






→ launch Instance

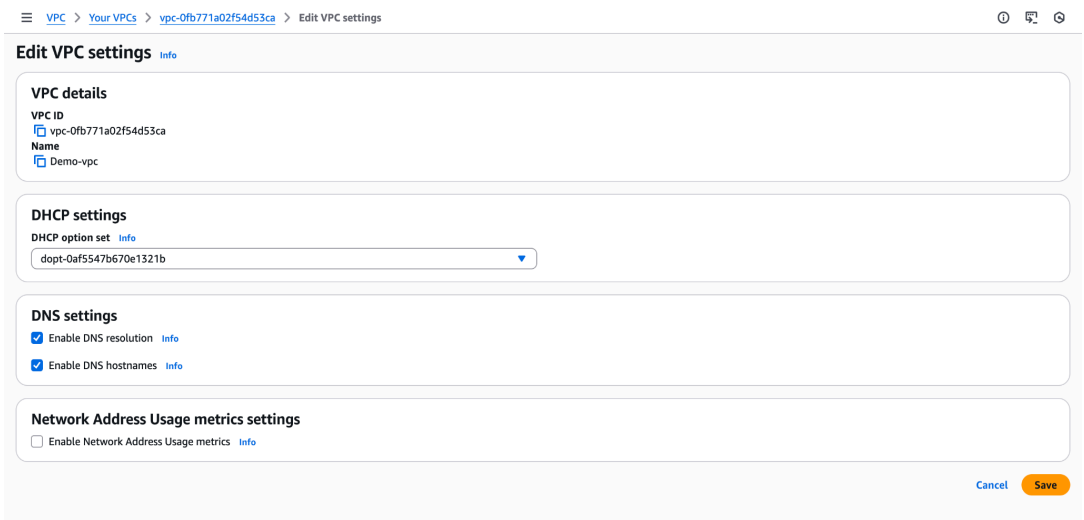
o/p:



Step 7: launch another EC2 Instance for private EC2

→ enable the DNS in vpc

Go to demo-vpc→ actions → select edit vpc settings → select enable DNS hostname (save it)



Ec2 instance name:

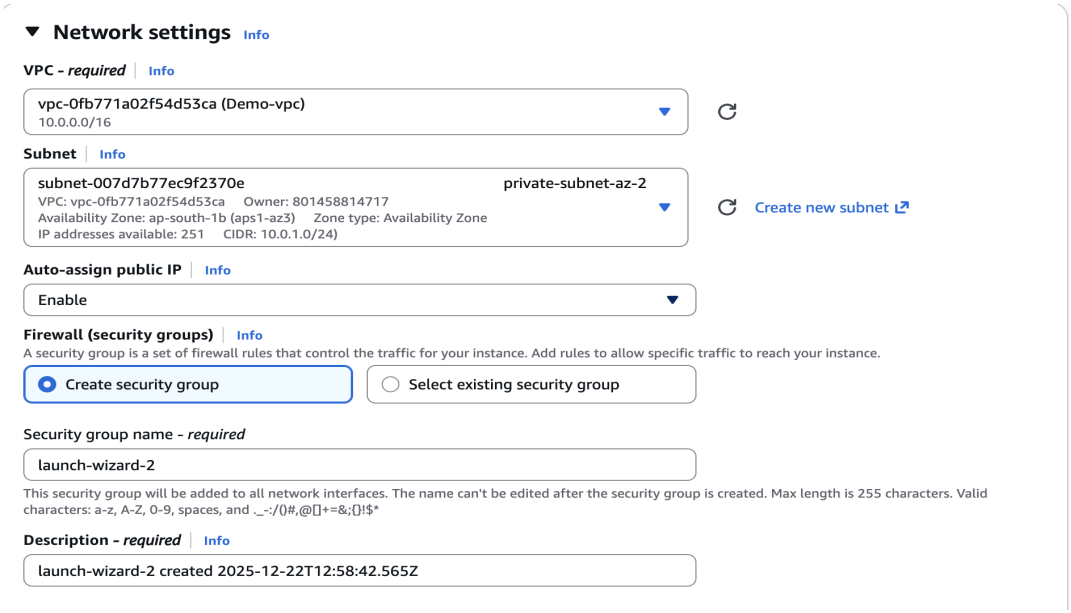
Name : Private EC2

AMI: amazon linux

Instance type: t2.micro

Key pair: select old one

Network settings:



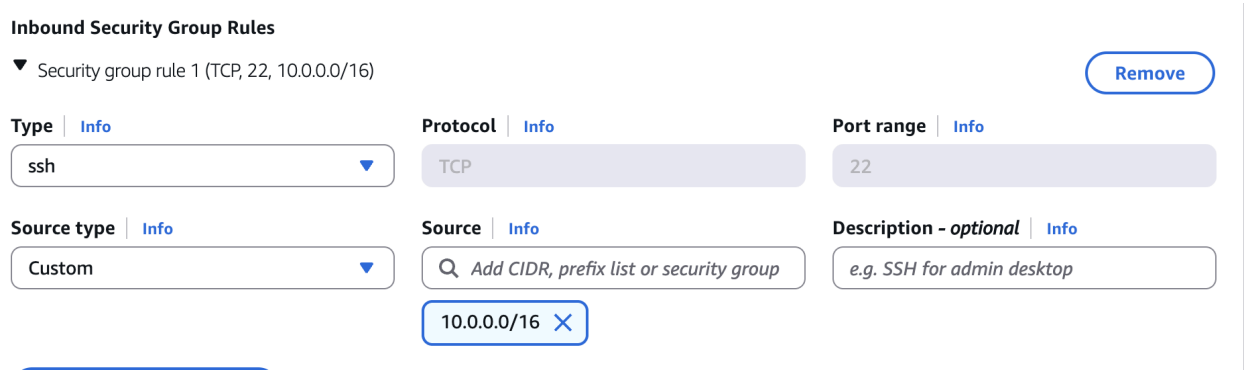
Security group:

type : ssh

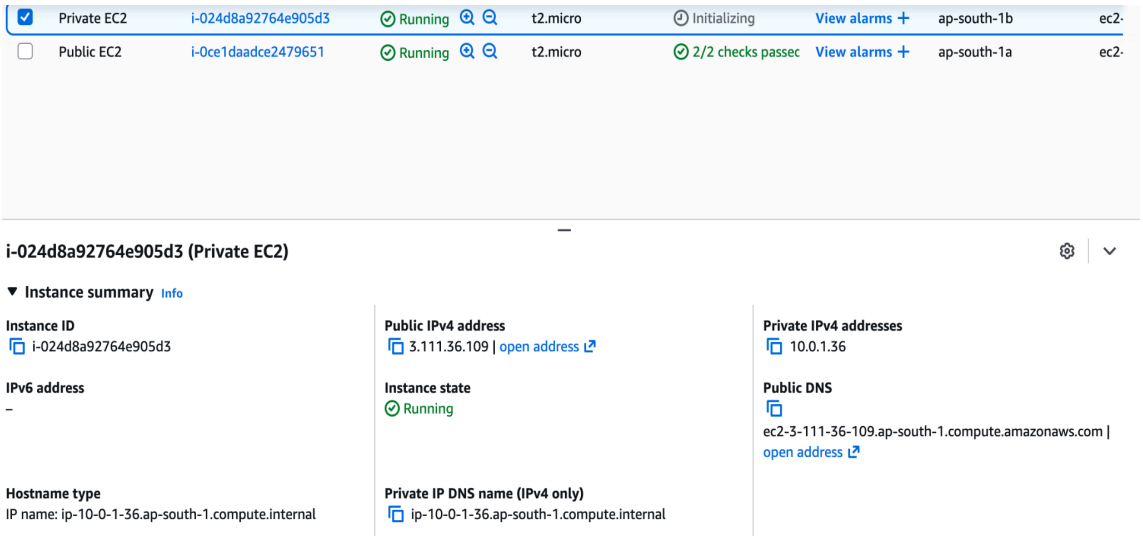
Protocol :TCP

Port range:22

Source : custom (10.0.0.0/16) (By setting the security group source to 10.0.0.0/16, the EC2 instance allows connections only from servers inside the same VPC. Any connection from outside the VPC or from the internet is blocked)



o/p:we got public and private dns names .to access the server we can use DNS names instead of IP addresses



Step 8: connect public ec2 instance to local

Go to terminal

> cd downloads

>chmod 400 "newly.pem"

>ssh -i "newly.pem" ec2-user@ec2-15-206-81-22.ap-south-1.compute.amazonaws.com

```
durgasanthoshi@Durgas-MacBook-Air downloads % chmod 400 "newly.pem"
durgasanthoshi@Durgas-MacBook-Air downloads % ssh -i "newly.pem" ec2-user@ec2-15-206-81-22.ap-south-
1.compute.amazonaws.com
The authenticity of host 'ec2-15-206-81-22.ap-south-1.compute.amazonaws.com (15.206.81.22)' can't be
 established.
ED25519 key fingerprint is SHA256:fFC/qENomEbTaWdx1ksbtNkwjNeVjTgDlYJvIDbOh8Y.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-15-206-81-22.ap-south-1.compute.amazonaws.com' (ED25519) to the list
 of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-0-232 ~]$
```

Step 9: now try to connect private ec2 instance to local

```
durgasanthoshi@Durgas-MacBook-Air ~ % cd downloads
durgasanthoshi@Durgas-MacBook-Air downloads % chmod 400 "newly.pem"
durgasanthoshi@Durgas-MacBook-Air downloads % ssh -i "newly.pem" ec2-user@ec2-3-111-36-109.ap-south-
1.compute.amazonaws.com
```

o/p: operation time out error . because private ec2 instance is not having the internet connection

```
ssh: connect to host ec2-3-111-36-109.ap-south-1.compute.amazonaws.com port 22: Operation timed out
durgasanthoshi@Durgas-MacBook-Air downloads %
durgasanthoshi@Durgas-MacBook-Air downloads %
```

Step 10: now to try to connect public ec2 instance to private ec2 instance

→we have our key file in local downloads folder. Now we need to copy the key file.prem from your laptop to the public EC2.This allows public EC2 to connect to private EC2

Open a new terminal

>cd downloads

>ls (check the .pem file is there or not)

>scp -i newly.pem newly.pem ec2-user@15.206.81.22:/home/ec2-user/

(scp → copy a file this command copies my key file from my laptop to the public EC2)

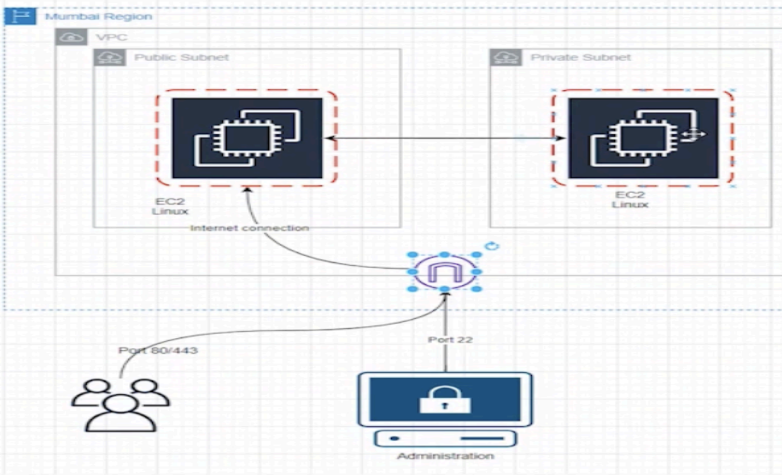
```
durgasanthoshi@Durgas-MacBook-Air ~ % cd downloads
durgasanthoshi@Durgas-MacBook-Air downloads % ls
ALB.pem                         DevOps Complete Roadmap.pdf
aws load balancer               linux-vm-key-6.pem
AWS Roadmap.pdf                 newly.pem
aws tasks                       Pan card.jpeg
Bio data                        Radha krishna.jpeg
BIO DATA.pdf                    s3
Biodata.pdf                     test3_credentials.csv
durgasanthoshi@Durgas-MacBook-Air downloads % scp -i newly.pem newly.pem ec2-use
r@<PUBLIC-EC2-PUBLIC-IP>:/home/ec2-user/

zsh: no such file or directory: PUBLIC-EC2-PUBLIC-IP
durgasanthoshi@Durgas-MacBook-Air downloads % scp -i newly.pem newly.pem ec2-use
r@15.206.81.22:/home/ec2-user/
The authenticity of host '15.206.81.22 (15.206.81.22)' can't be established.
ED25519 key fingerprint is SHA256:fFC/qENomEbTaWdx1ksbtNkwjNeVjTgDlYJvIDbOh8Y.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:40: ec2-15-206-81-22.ap-south-1.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '15.206.81.22' (ED25519) to the list of known hosts.
newly.pem                                     100% 1674    51.1KB/s   00:00
durgasanthoshi@Durgas-MacBook-Air downloads %
```

—> now we have access to connect private ec2 instance in public ec2 instance

>copy the private ip address from the ec2 instance

```
[ec2-user@ip-10-0-0-232 ~]$ ssh -i "newly.pem" ec2-user@ec2-3-111-36-109.ap-south-1.compute.amazonaw
s.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-1-36 ~]$
```

Flow chat :




```
       _/m/'
[ec2-user@ip-10-0-1-36 ~]$ logout
Connection to ec2-3-111-36-109.ap-south-1.compute.amazonaws.com closed.
[ec2-user@ip-10-0-0-232 ~]$
```

> logout from the Private ec2 instance in public ec2 instance

Step 11: **run any web server in public ec2 instance**

>install apache server in public ec2 instance

Cmd:

>sudo yum install httpd -y

>sudo systemctl start httpd

>sudo nano /var/www/html/index.html

Paste the code

<!DOCTYPE html>

<html>

<head>

<title>AWS VPC Public EC2</title>

</head>

<body>

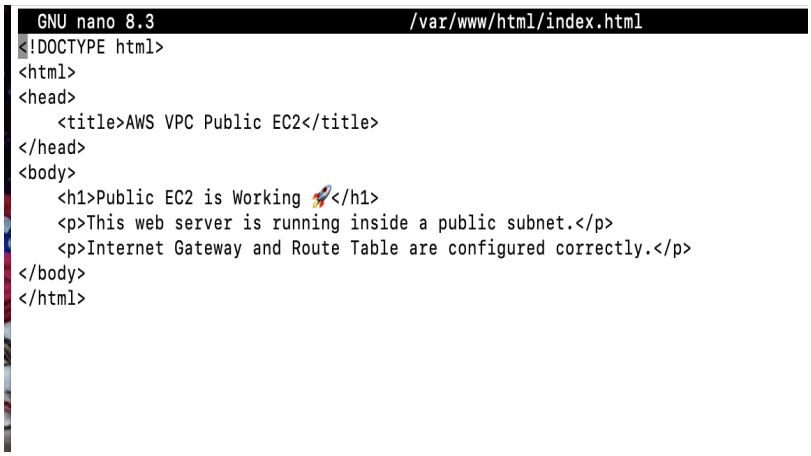
<h1>Public EC2 is Working 🚀</h1>

This web server is running inside a public subnet.

Internet Gateway and Route Table are configured correctly.

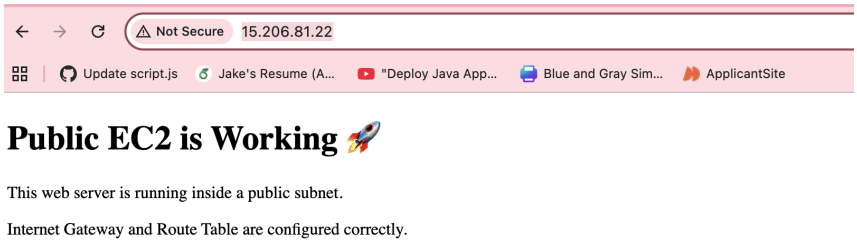
</body>

</html>



→save the code
→http://<PUBLIC-EC2-PUBLIC-IP> (paste the public ec2 ipv4 address) (ex:http://15.206.81.22/)

o/p:



Finally our web page is running successfully

Summary of what i did:

Reference video :https://www.youtube.com/watch?v=cyy2ok-00qs

- Created **VPC**
- Created **Public & Private subnets**
- Attached **Internet Gateway**
- Configured **Route Tables**
- Launched **Public EC2 & Private EC2**
- Connected **Public → Private EC2**
- Hosted a **web page on Public EC2**

created a VPC with public and private subnets.

The public EC2 is connected to the internet and hosts a web page.

The private EC2 has no internet and can only be accessed securely via the public EC2.

I set up proper route tables, Internet Gateway, and security groups to control access.

Security
Private EC2 is not exposed to the internet
Only accessible from public EC2

Controlled access
Public EC2 acts as a **jump server**
Allows safe internal communication

Can add more private servers without giving them internet